DATA ACQUISITION

Circuit Puts Analog Data into Excel

Clayton B. Grantham
Burr-Brown Corp., Tucson, AZ

A few integrated circuits, a simple Excel 5 macro, and a driver are all you need to transform a PC into a virtual voltmeter. The resulting 16-bit voltmeter circuit offers an input range of up to ±200 V, resolves to 6.1 mV, and has an accuracy of 0.005% after calibration.

The interface circuit (**Fig. 1**) that connects the 16-bit ADS7807 analog-to-digital converter (ADC) IC to a PC's parallel port is simple. The control software forces nine of the port's signal lines to a high state, which lets the corresponding pins provide power to the circuit through 10-Ω resistors. Using the parallel interface's data lines to power the interface may not work for all PCs, though. If your PC cannot supply clean 4.75-V to 5.25-V DC power to the circuit, use an external source to supply the 35 mW the circuit needs. (For more information about the parallel port, see the IEEE 1284 standard,[1] which provides a thorough description of signals, controls, and hardware for a PC's parallel interface.)

The 74HCT04 inverter in the interface circuit provides a clean signal to a monostable (10-kΩ resistor and 100-pF capacitor) that produces a 150-ns logic-zero pulse for the ADC's R/C* input. When the R/C* input goes to a logic zero, the ADC starts a conversion. The short pulse ensures that the R/C* input is back at logic one before the ADC's BUSY* (conversion complete) signal goes to a logic one. The short pulse width also minimizes any digital feedthrough that might occur during a conversion.

Select a Byte

The 16-bit ADC's BYTE input selects the byte that appears at the ADC's outputs. In turn, the 74HCT157 multiplexer selects which nibble (four bits) of each byte to transmit to the parallel port. The macro controls the flow of four nibbles to the PC and reconstructs them into a 16-bit integer.

By itself, the ADS7807 has an input voltage range of ±10 V. Using an 866-kΩ input resistor, however, extends the range to ±200 V according to the equation:

$$V_{in(max)} = \pm 10\ V * (1 + R_{ext}/45.5k)$$

To reduce noise on your signal, place a capacitor between the ADS7807's input and ground to form an RC network with the input resistor.

To acquire voltages, select a cell or a range of cells in a spreadsheet and then execute the macro (**Fig. 2**, p. 14),

(text continued on p. 16)

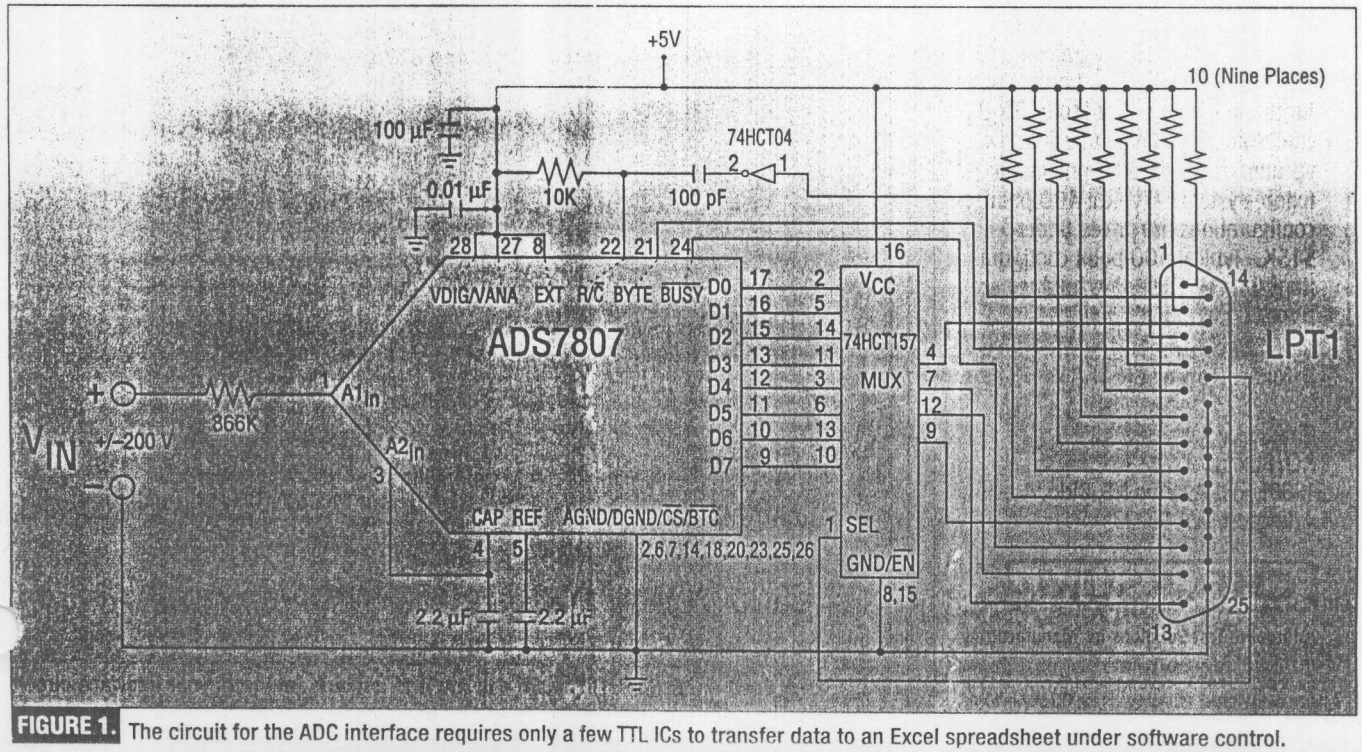

FIGURE 1. The circuit for the ADC interface requires only a few TTL ICs to transfer data to an Excel spreadsheet under software control.

```
'The following macro communicates through the PC's parallel printer port
'with the dynamic-link-library, CUSER3.DLL.
'Parallel port initalization

Declare Sub Out Lib "CUSER3.DLL" (ByVal Addr%, ByVal Byte%)
Declare Function Inp Lib "CUSER3.DLL" (ByVal Addr%) As Integer

Sub ADSinput()

Dim Datum As Integer                    'Temporary data element

'Hexadecimal Port Addresses
DataPortAddr = &h378
StatusPortAddr = &h379
ControlPortAddr = &h37a

If TypeName(Selection) <> "Range" Then Exit Sub
Out ControlPortAddr, &h4                'Initialize control port
Out DataPortAddr, &hff                  'Power up ADS7807
For i = 1 To Selection.Count
    Out ControlPortAddr, &h6            'Start Conversion
    For j = 1 To 2                      'Wait for notBusy signal
    Next j                              'Not needed for slow PCs
    If (Inp(StatusPortAddr) And &h80) Then
        Selection.Item(1).Value = "HDErr: Hardware error."
        Exit Sub
    End If

    'Input low byte, high nibble. Isolate & shift nibble up
        Datum = ((Inp(StatusPortAddr) And &h78) * &h2)

        Out ControlPortAddr, &hc 'Select low byte, high nibble
        'Input low byte, low nibble. Add in nibble and place in cell
        Datum = Datum + (Inp(StatusPortAddr) And &h78) / &h8

        Selection.Item(i).Value = Datum

        Out ControlPortAddr, &h0 'Select high byte, high nibble

        'Input high byte, high nibble. Isolate nibble
        Datum = (Inp(StatusPortAddr) And &h78)

        If (Datum And &h40) Then
                                    'Test MSB of nibble for negative value
                                    'Scale negative values & shift nibble to top of word
        Datum = -32768 + (Datum And &h38) * &h200
        Else Datum = Datum * &h200          'Shift nibble to top of word
        End If

        Out ControlPortAddr, &h8            'Select high byte, low nibble

        'Input high byte, low nibble. Isolate & shift nibble up & add to word
        Datum = Datum + ((Inp(StatusPortAddr) And &h78) * &h20)

        'Convert result to signed floating point
        Selection.Item(i).Value = (Datum + Selection.Item(i).Value) / 3276.8

        'Scale result with offset and gain calfactors for +/- 200V input
        Selection.Item(i).Value = (Selection.Item(i).Value - 1.712) * 19.704

        Out ControlPortAddr, &h4            'Reset for another conversion
        Next i
End Sub
```

FIGURE 2. Load the Excel Basic macro and run it to acquire data from your interface circui
Be sure the hexadecimal port addresses match the addresses on the port you're using.

```
Sub SoftCal()
        Set B1 = Worksheets("MyWorksheet").Range("B1")
        Set B2 = Worksheets("MyWorksheet").Range("B2")
        Set B3 = Worksheets("MyWorksheet").Range("B3")
        Set B4 = Worksheets("MyWorksheet").Range("B4")
        Message = "Connect CAL source to Input; Enter CAL Value"
        FirstCalValue = Val(InputBox(Message))
        B1.Select
        ADSinput
        Message = "Change CAL source; Enter CAL Value"
        SecondCalValue = Val(InputBox(Message))
        B2.Select
        ADSinput
        B3.Value = (FirstCalValue – SecondCalValue) / (B1.Value – B2.Value)
    'Gain coefficient.
        B4.Value = FirstCalValue / B3.Value – B1.Value
    'Offset coefficient.
End Sub
```

FIGURE 3. If you don't want to use hardware trimmers, use this short macro to obtain gain and offset values.

which contiguously fills each cell with a measurement. The Excel macro uses a dynamic link library routine, CUSER3.DLL, to communicate with the ADC through the PC's parallel port. (*T&MW* provides the DLL file for free; see "Notes," this page.) Be sure to transfer the CUSER3.DLL into your main Excel directory so the macro can locate it.

Control Your PC's Ports

To keep Excel Basic—the Basic-like language built into Excel—independent of hardware, Microsoft excluded commands that control specific I/O ports. The CUSER3.DLL circumvents this limitation and lets you transfer bytes to output ports and retrieve bytes from input ports.

Typically, a PC's parallel port resides at h0378 (data), h0379 (status), and h037A (control). Check your PC's manual to be sure you have the proper addresses for the port you want to use. If necessary, change the three address definitions at the start of the macro.

When you run the macro, it initializes the control byte and starts the ADC. After a delay of at least 25 µs, the macro checks the ADC's BUSY signal. At the end of a conversion, the macro inputs the ADC's data. For example, if you select 15 cells, the macro acquires 15 readings, one after the other. The timing between conversions depends on your PC's timing and the timing overhead of Excel. My 33-MHz 486-based PC took 23 ms per conversion, and the Nyquist bandwidth was 22 Hz.

Calculate Offset and Gain Errors

The basic circuit doesn't compensate for gain and offset errors. You can use nominal ADS7807 offset and gain errors (without external trimming) to scale the results obtained with the macro. Use the code shown in **Fig. 3** with the ADSinput() macro to provide gain and offset coefficients, if you need them. To achieve the highest accuracy, add external trimmer resistors to the circuit.[2] *T&MW*

FOOTNOTES

1. IEEE 1284-1994, *IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers,* IEEE, Piscataway, NJ.

2. *ADS7807 Low-Power 16-Bit Sampling CMOS Analog-to-Digital Converter,* data sheet Burr-Brown, Tucson, AZ. 1992.

NOTES

The DLL and macro files are available through e-mail. Send a message to *tmw@cahners.com* with only CUSER3 in the subject field. In addition to the DLL, you'll receive a DLL source file, a DLL documentation file, and a text file of the two macros. You can also mail us a blank, PC-formatted, high-density 3½-in. disk and your mailing address.

Clayton B. Grantham *is the Linear Div. product engineering manager at Burr-Brown Corp. He has B.S.E.E. and M.S.E.E. degrees from the University of Arizona.*